Locally based kernel PLS regression de-noising with application to event-related potentials

Roman Rosipal[1,2]
Leonard J. Trejo[1]
Kevin Wheeler[1]
[1]*NASA Ames Research Center*
Computational Sciences Division, Moffett Field , CA
[2]*Department of Theoretical Methods*
Slovak Academy of Sciences, Bratislava, Slovak Republic
Peter Tiňo
Neural Computing Research Group
Aston University, Birmingham, UK

Abstract

The close relation of signal de-noising and regression problems dealing with the estimation of functions reflecting dependency between a set of inputs and dependent outputs corrupted with some level of noise have been employed in our approach. In signal processing desired functions (signals) are usually assumed to be a linear combination of the basis functions $\phi_i(\mathbf{x})$; i.e. :

$$f(\mathbf{x}) = \sum_{d=1}^{D} w_d \phi_d(\mathbf{x}) + w_0 \ . \tag{1}$$

With respect to this signal de-noising formulation our method consist from the following steps used with the aim of the function $f(\mathbf{x})$ recovery from the original noisy signal measurement:

- inputs ($\mathbf{x}$) are equidistantly sampled points in input space; in 1-D we pre-define sampling interval to be $[-1, 1]$ and the number of sampling points then depends on the selected sampling rate. This allows us to find optimal or near optimal parameters for the kernel mapping (or even particular kernel mapping) for different classes of signals under investigation.

- the basis function $\phi_i(\mathbf{x})$ are taken to be components obtain by kernel PLS, which may be seen as the estimates of orthogonal basis in a feature space defined by kernel function used. These estimates are sequentially obtained using the existing correlations between nonlinearly mapped input data and the measured noisy signal [1].

- to set the number of basis functions D we have used the VC-based model selection criterion described in [2,3,4]. The ordering of the basis functions for the purposes of the used model selection criterion is defined by their sequential extraction.

- using the locally based kernel PLS allows us to deal with a possible discontinuity and non-stationarity in the signal of interest. Locality is achieved using modified kernel PLS algorithm incorporating the weight functions reflecting the local areas of interest. Depending on weight function selection this allows us to construct soft or hard thresholding regions where kernel PLS regression models are constructed. Final estimate consist of composition of individual local kernel PLS regression models.

- we compared our methodology with the state-of-the-art wavelet based signal de-noising and smoothing splines approaches on heavisine and artificially generated event-related potentials distributed over individual scalp areas. Different levels of additive white and colored noises with respect to clean signals were used.

Kernel PLS regression

- linear PLS regression in feature space $\mathcal{F}$
- decomposition: $\mathbf{X} = \mathbf{T}\mathbf{P}^T + \mathbf{E}$; $\mathbf{Y} = \mathbf{U}\mathbf{C}^T + \mathbf{F}$
- latent variables (scores):
  $\mathbf{t}_i = \sum_{k=1}^{K} w_{ik}(\mathbf{x}_k - \sum_{b=1}^{i-1} \mathbf{t}_b p_{bk})$
  $\mathbf{u}_i = f(\mathbf{t}_i) + \mathbf{h}_i$ – inner relation in PLS model;
  $\mathbf{h}_i$ - vector of residuals
- NIPALS algorithm applied to PLS finds weights $\mathbf{w}, \mathbf{c}$ such that
  $[cov(\mathbf{t}, \mathbf{u})]^2 = [cov(\mathbf{X}\mathbf{w}, \mathbf{Y}\mathbf{c})]^2 = max_{|\mathbf{r}|=|\mathbf{s}|=1}[cov(\mathbf{X}\mathbf{r}, \mathbf{Y}\mathbf{s})]^2$
- nonlinear (kernel) variant [1]:
  $\mathbf{X}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{t} = \mathbf{K}\mathbf{Y}\mathbf{Y}^T\mathbf{t} = \lambda\mathbf{t}$
  $\mathbf{u} = \mathbf{Y}\mathbf{Y}^T\mathbf{t}$
  or iterative kernel-based NIPALS algorithm
- sequential extraction $\mathbf{t}, \mathbf{u} \Rightarrow \mathbf{T}, \mathbf{U}$
- deflation of $\mathbf{K}$ and $\mathbf{Y}$ matrices after each step
- final regression model:

$$\hat{\mathbf{Y}} = \mathbf{X}\mathbf{B} = \mathbf{K}\mathbf{U}(\mathbf{T}^T\mathbf{K}\mathbf{U})^{-1}\mathbf{T}^T\mathbf{Y} = \mathbf{T}\mathbf{T}^T\mathbf{Y} = \mathbf{T}\tilde{\mathbf{B}}$$

assuming $y \in \mathcal{R}$

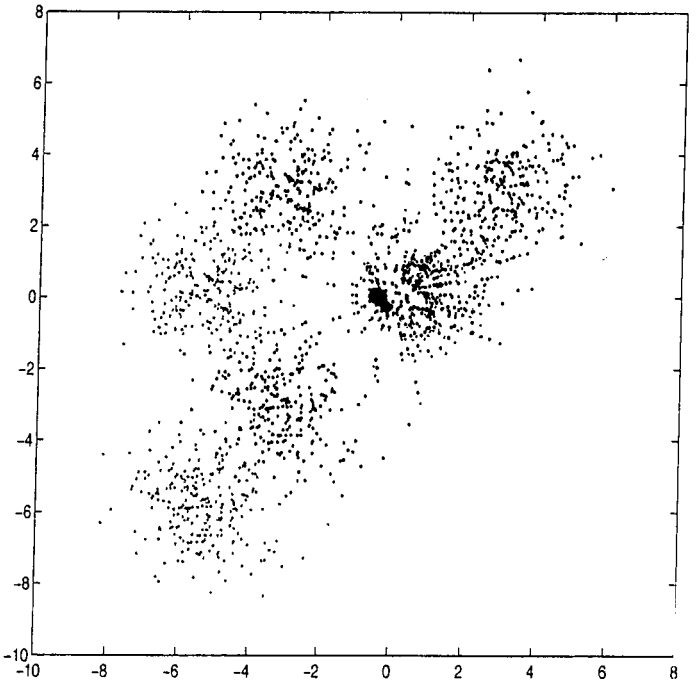
$$\hat{y}(\mathbf{x}) = \tilde{b}_1 t_1(\mathbf{x}) + \tilde{b}_2 t_2(\mathbf{x}) + \ldots + \tilde{b}_D t_D(\mathbf{x}) \quad ; \quad \tilde{\mathbf{b}} = \mathbf{T}^T\mathbf{y}$$

Locally based kernel PLS regression

- soft clustering : $\mathbf{r}$ - vector of weights

$$r_s = \sum \mathbf{r} \; ; \; \mathbf{R}_d = diag(\mathbf{r}) \; ; \; \mathbf{J} = ones(n, 1) \; ; \; \mathbf{I} = eye(n)$$

$$\mathbf{X}_r = \mathbf{R}_d(\mathbf{X} - \mathbf{J}\frac{\mathbf{r}^T\mathbf{X}}{r_s}) \;\; ; \;\; mean(\mathbf{X}_r) = \mathbf{0}$$



- kernel variant:

$$\mathbf{K}_r = \mathbf{X}_r\mathbf{X}_r^T = \mathbf{R}_d(\mathbf{I} - \frac{\mathbf{J}\mathbf{r}^T}{r_s})\mathbf{K}(\mathbf{I} - \frac{\mathbf{J}\mathbf{r}^T}{r_s})^T\mathbf{R}_d$$

$$\mathbf{Y}_r = \mathbf{R}_d(\mathbf{Y} - \mathbf{J}\frac{\mathbf{r}^T\mathbf{Y}}{r_s})$$

- Gaussian kernel:

$$\|\mathbf{x}_1 - \mathbf{x}_2\|^2 = \langle \mathbf{x}_1, \mathbf{x}_1 \rangle - 2\langle \mathbf{x}_1, \mathbf{x}_2 \rangle + \langle \mathbf{x}_2, \mathbf{x}_2 \rangle \Rightarrow$$

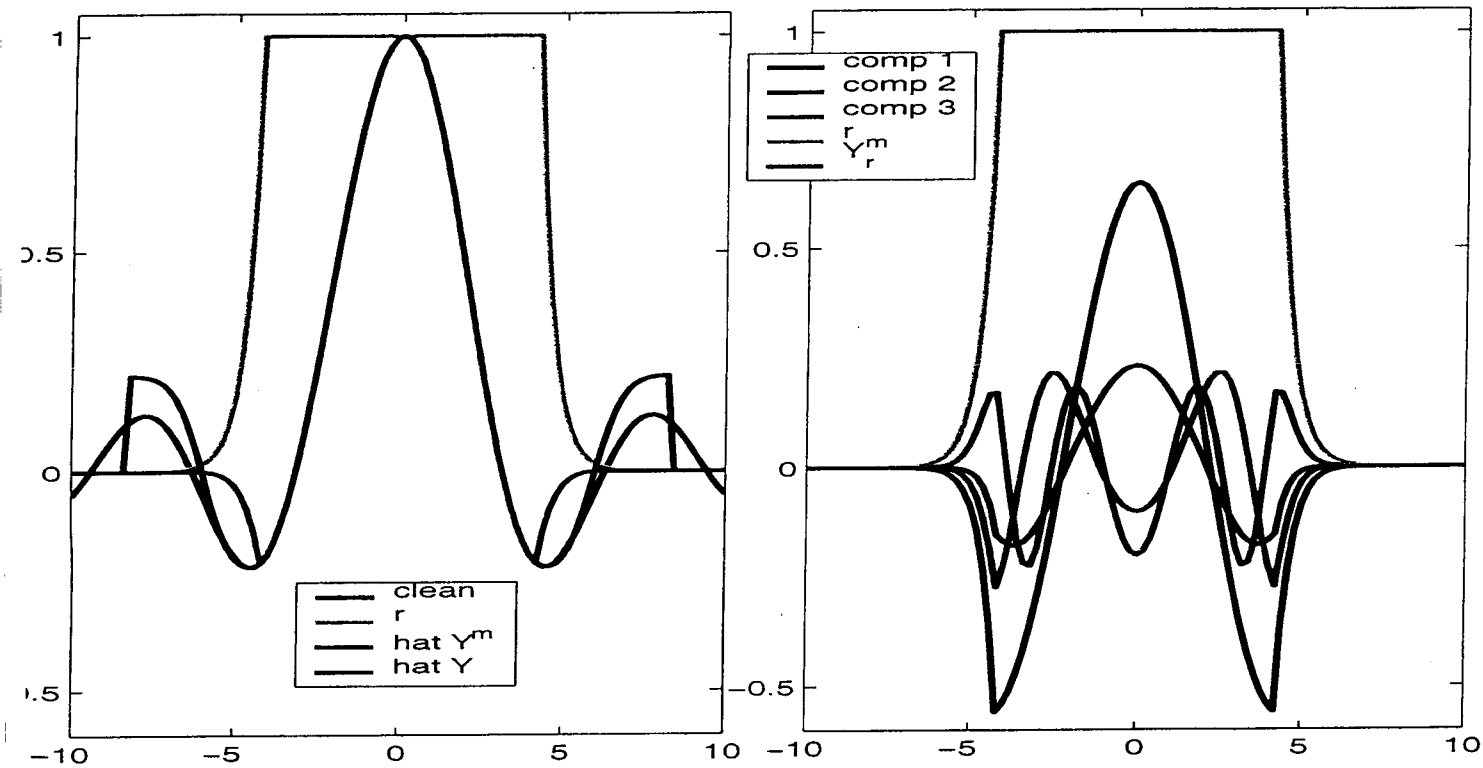
$$\|\psi(\mathbf{x}_1) - \psi(\mathbf{x}_2)\|^2 = 2 - 2\exp(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{h})$$

- local kernel PLS regression
  m-th cluster defined by weight vector $\mathbf{r}^m$:
  sequential extraction of $\mathbf{t}, \mathbf{u}$ from
  $\mathbf{K}_r^m \mathbf{Y}_r^m (\mathbf{Y}_r^m)^T$; $\mathbf{Y}_r^m (\mathbf{Y}_r^m)^T \Rightarrow \mathbf{T}^m, \mathbf{U}^m$

$$\hat{\mathbf{Y}}_r^m = \mathbf{T}^m (\mathbf{T}^m)^T \mathbf{Y}_r^m \Rightarrow \hat{\mathbf{Y}}^m = \mathbf{R}_d^{-1} \hat{\mathbf{Y}}_r^m + \mathbf{J} \frac{\mathbf{r}^T \mathbf{Y}}{r_s}$$

- final model (M - clusters):

$$\hat{Y}_i = \sum_{m=1}^{M} r_i^m \hat{Y}_i^m / \sum_{m=1}^{M} r_i^m \qquad i = 1, \ldots, n$$



VC-based complexity control

- for regression problems with squared loss the following bound on prediction risk (PR) holds with probability $1-\eta$ [4]

$$PR \leq \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 * (1-c\sqrt{\frac{h(\ln(\frac{an}{h})+1)-\ln\eta}{n}})_+^{-1}$$

h - VC dimension of the set of approximating functions
c - constant reflecting the "tails of the loss function distribution"
a - theoretical constant
$(x)_+ = u \; if \; x > 0$
$0 \; otherwise$

- Cherkassky et. al constructed empirical (heuristical) Vapnik's measure [2,3] to compute estimated risk (ER)

$$ER = \frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 * (1-\sqrt{b-b\ln b+\frac{\ln n}{2n}})_+^{-1}$$

$b=(d+1)/n$ where $d+1$ represents VC dimension of the approximation function (1) with d terms

Smoothing splines

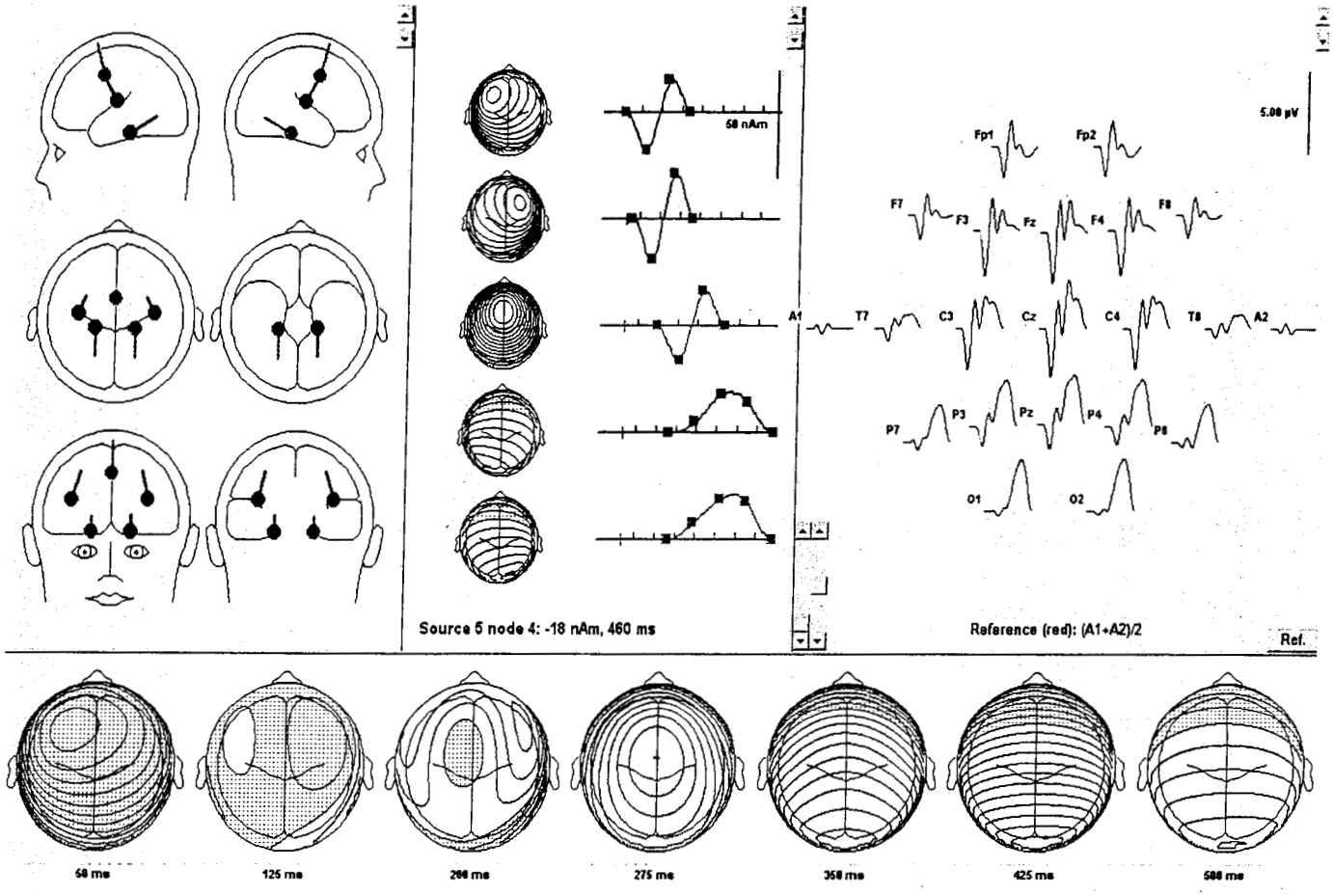
- $min_f(\frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda\int_a^b(f^{(2)}(x))^2dx \quad \lambda > 0 \Rightarrow$
  natural cubic splines with knots at x_i ; $i = 1, \ldots, n$
- $\hat{\mathbf{f}} = \mathbf{N}\mathbf{s}$; $\mathbf{s} = (\mathbf{N}^T\mathbf{N} + \lambda\mathbf{\Omega})^{-1}\mathbf{N}^T\mathbf{y}$
  $\{\mathbf{N}\}_{ji} = N_j(x_i)$; $j, k = 1, \ldots, n$
  $\{\mathbf{\Omega}\}_{jk} = \int_a^b N_j^{(2)}(t)N_k^{(2)}(t)dt$
- $\hat{\mathbf{f}} = \mathbf{S}_\lambda\mathbf{y} \rightarrow df_\lambda = trace(\mathbf{S}_\lambda)$
- squared loss
  $loo = \frac{1}{n}\sum_{i=1}^{n}[\frac{1 - \hat{f}(x_i)}{1 - (\mathbf{S}_\lambda)_{ii}}]^2 \rightarrow gcv = \frac{1}{n}\sum_{i=1}^{n}[\frac{1 - \hat{f}(x_i)}{1 - trace(\mathbf{S}_\lambda)/n}]^2$
- complete basis $\rightarrow$ *shrink* the coefficients toward smoothing

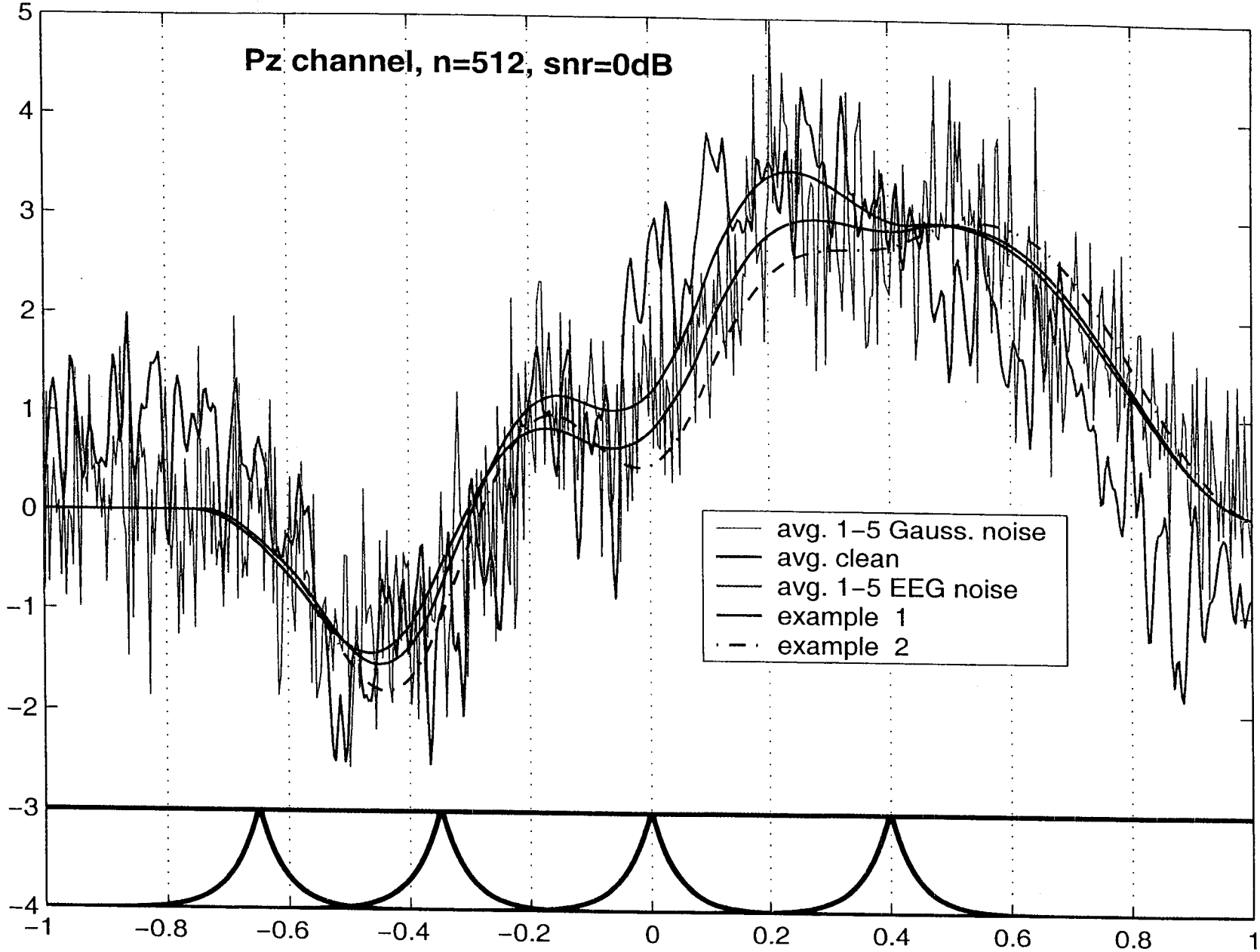
Wavelet smoothing

- complete orthonormal basis $\rightarrow$ *shrink* and *select* the coefficients toward a **sparse** representation
- wavelet basis is *localized in time and frequency*
- $\mathbf{y}^* = \mathbf{W}^T\mathbf{y}$; *discrete wavelet transform*
  (i.e. full LS regression coefficient)
  $\mathbf{W}_{n\times n}$ orthonormal basis
- *SURE shrinkage* : $\min_{\mathbf{r}} \|\mathbf{y} - \mathbf{W}\mathbf{r}\|_2^2 + 2\lambda\|\mathbf{r}\|_1 \Rightarrow$
  $\hat{r}_j = sign(y_j^*)(|y_j^*| - \lambda)_+$
  ?? $\lambda = \sigma\sqrt{2\log n}$
- $\hat{\mathbf{f}} = \mathbf{W}\hat{\mathbf{r}}$ *inverse wavelet transform*

Data construction

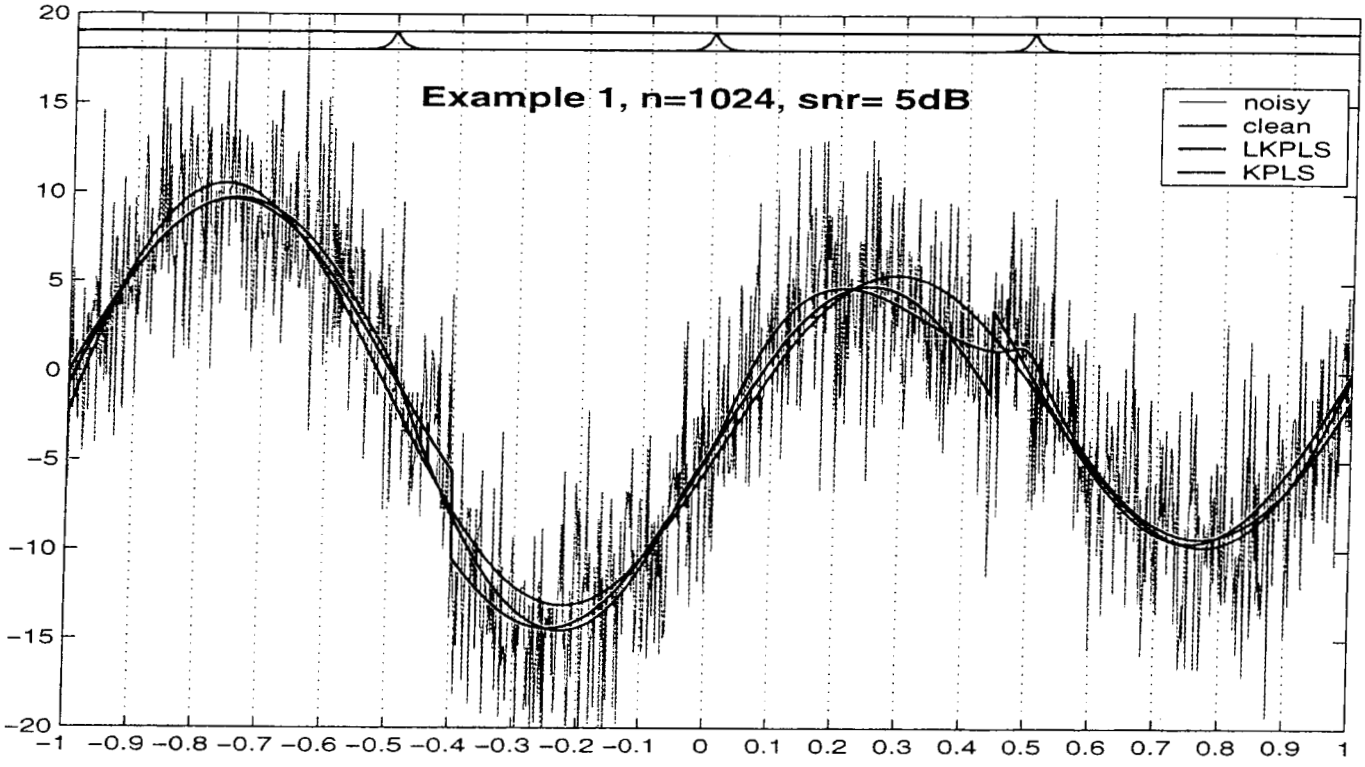
- heavisine function
  additive noise: white Gaussian
- Event related potentials - N100,P300
  additive noise : white Gaussian,
  relax state spatially distributed EEG signal




Pz channel, n=512, snr=0dB
avg. 1–5 Gauss. noise
avg. clean
avg. 1–5 EEG noise
example 1
example 2
5
4
3
2
1
0
−1
−2
−3
−4
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1

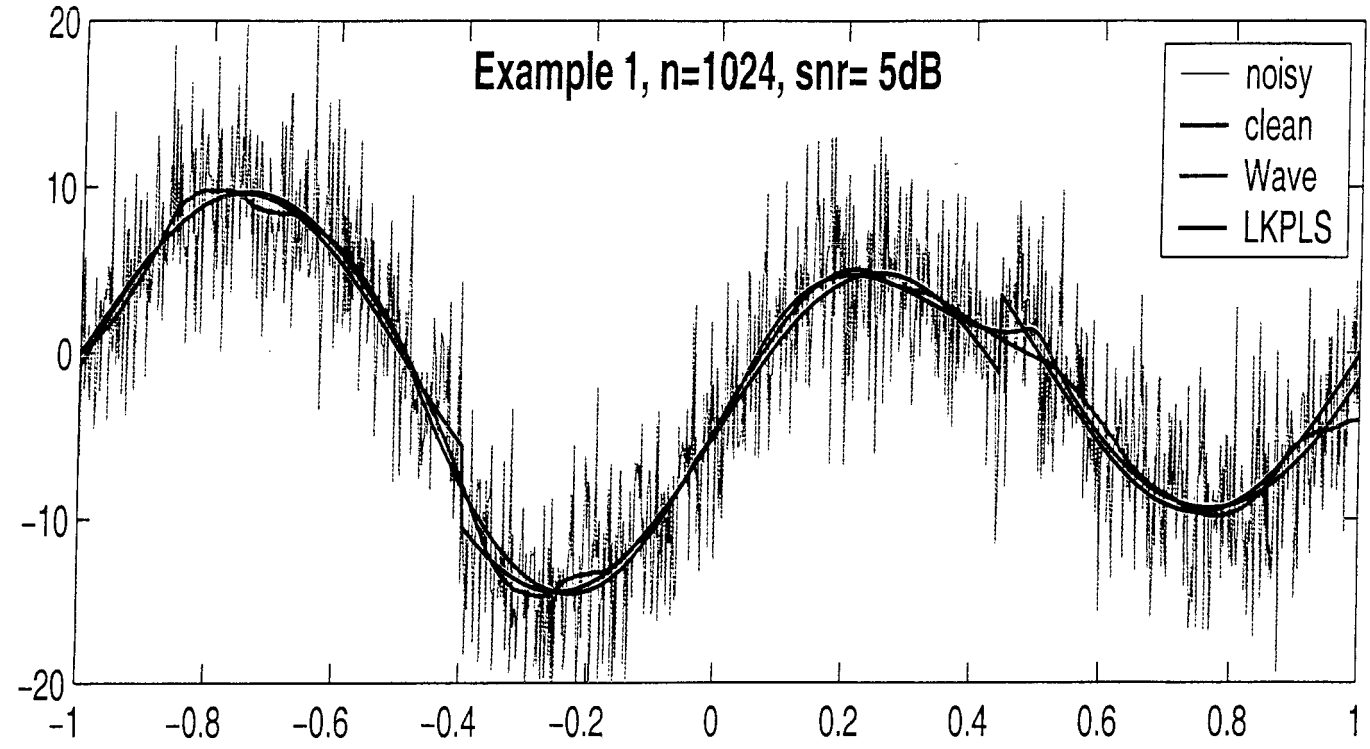
Results

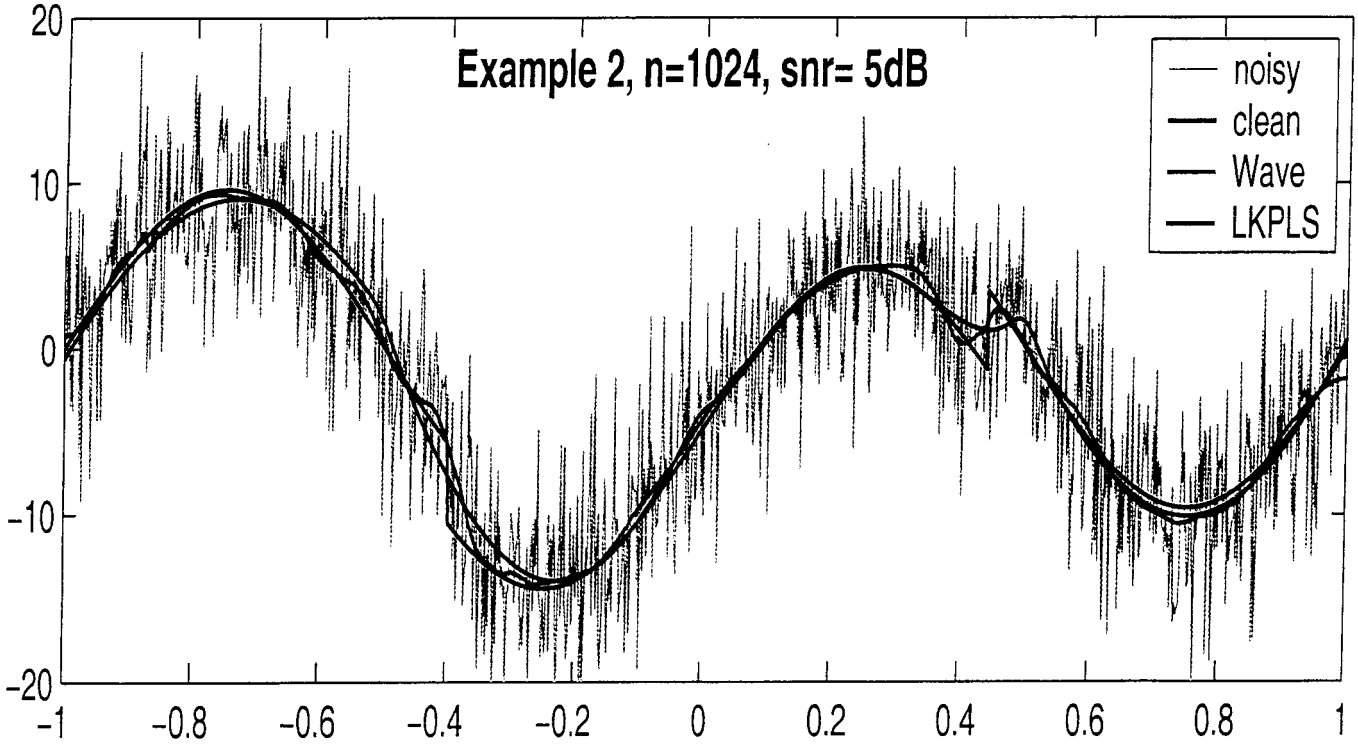
- heavisine function



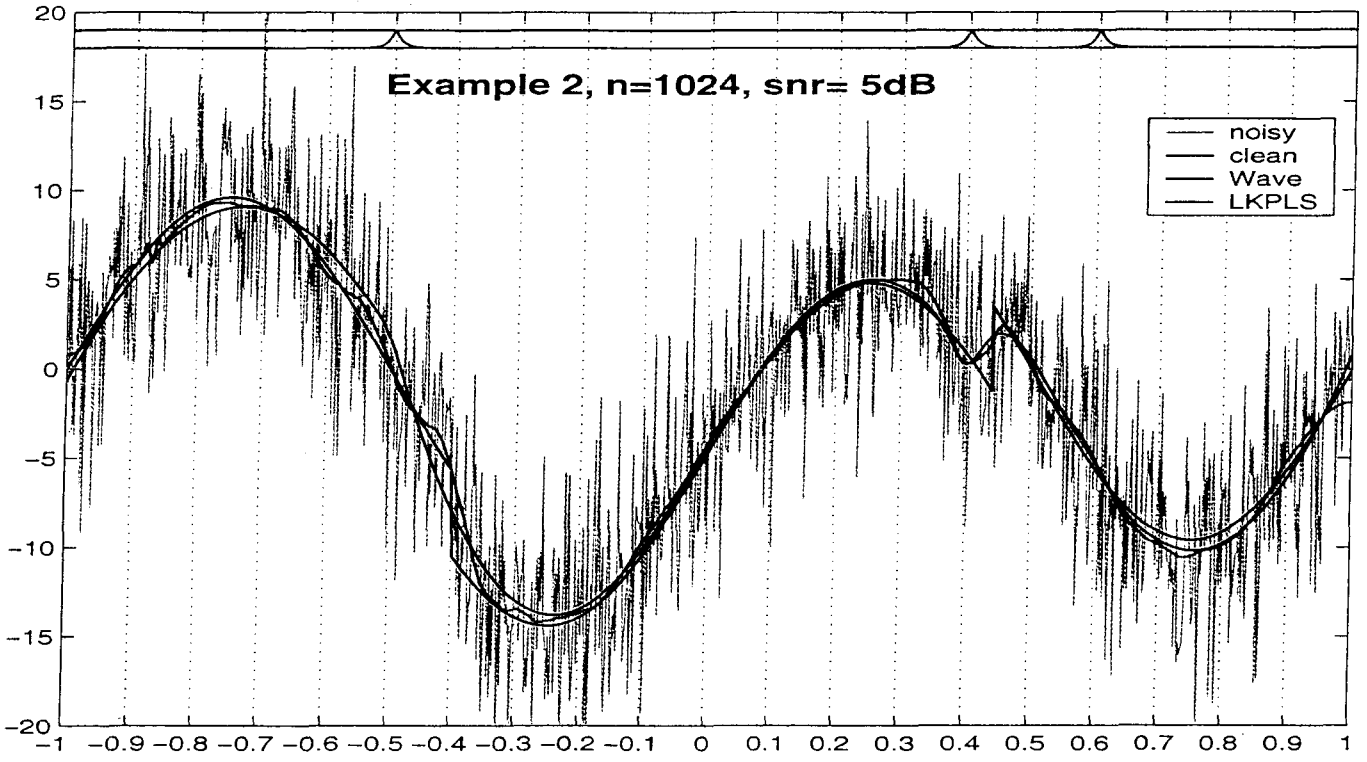
Noise	SNR 1dB		SNR 5dB		SNR 10dB		SNR 15dB	
	LKPLS	Wave	LKPLS	Wave	LKPLS	Wave	LKPLS	Wave
128	.32	.33	.21	.22	.14	.14	.10	.10
256	.23	.25	.16	.17	.11	.12	.08	.08
512	.17	.19	.13	.14	.10	.10	.07	.07
1024	.13	.15	.11	.11	.08	.08	.07	.05
2048	.12	.12	.10	.09	.07	.06	.07	.04

Table 1: Normalized root mean squared error.


Example 1, n=1024, snr= 5dB
noisy
clean
Wave
LKPLS
20
10
0
-10
-20
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


Example 2, n=1024, snr= 5dB
noisy
clean
Wave
LKPLS
20
10
0
-10
-20
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

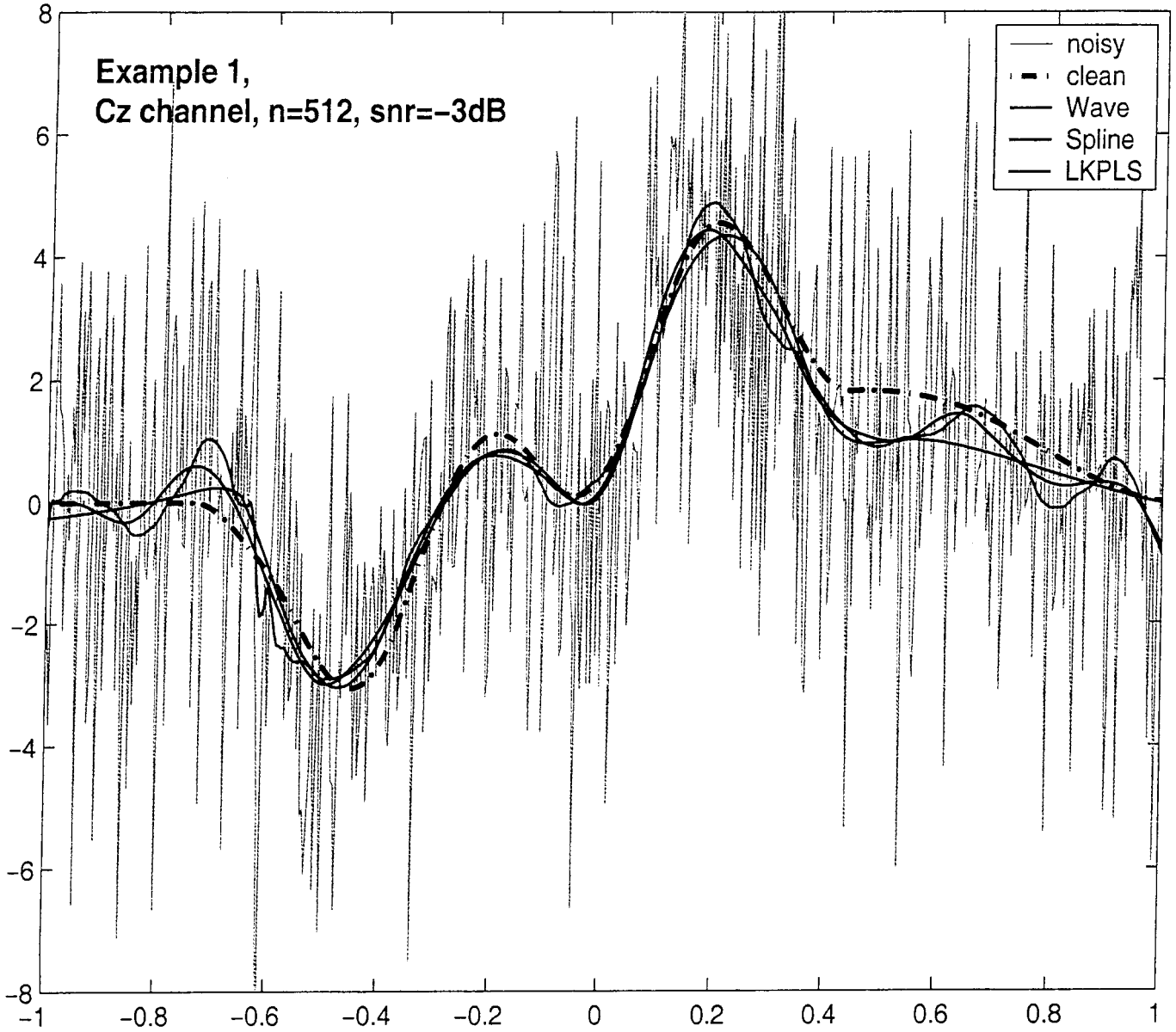
Let's cheat - a little bit !!

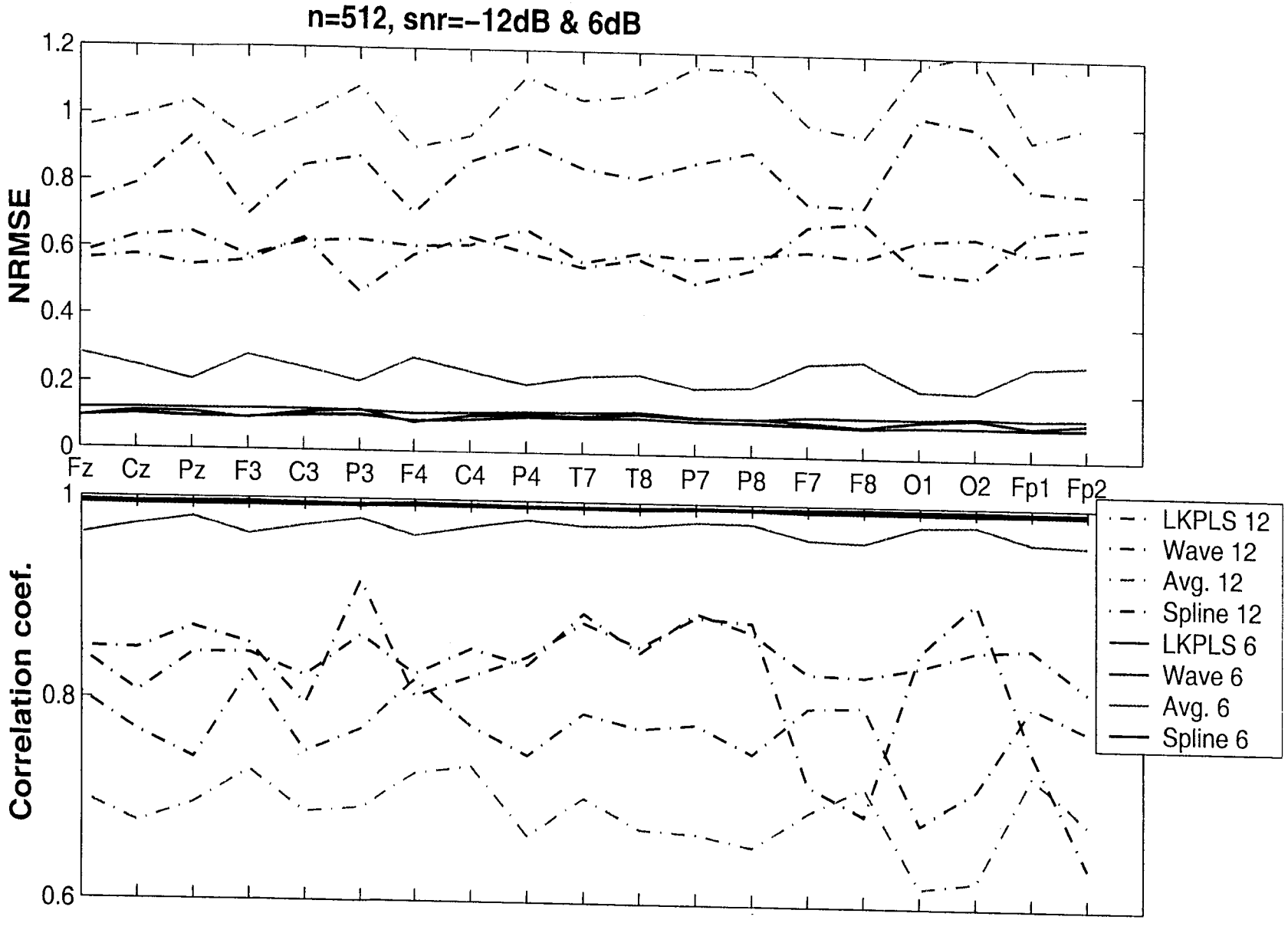


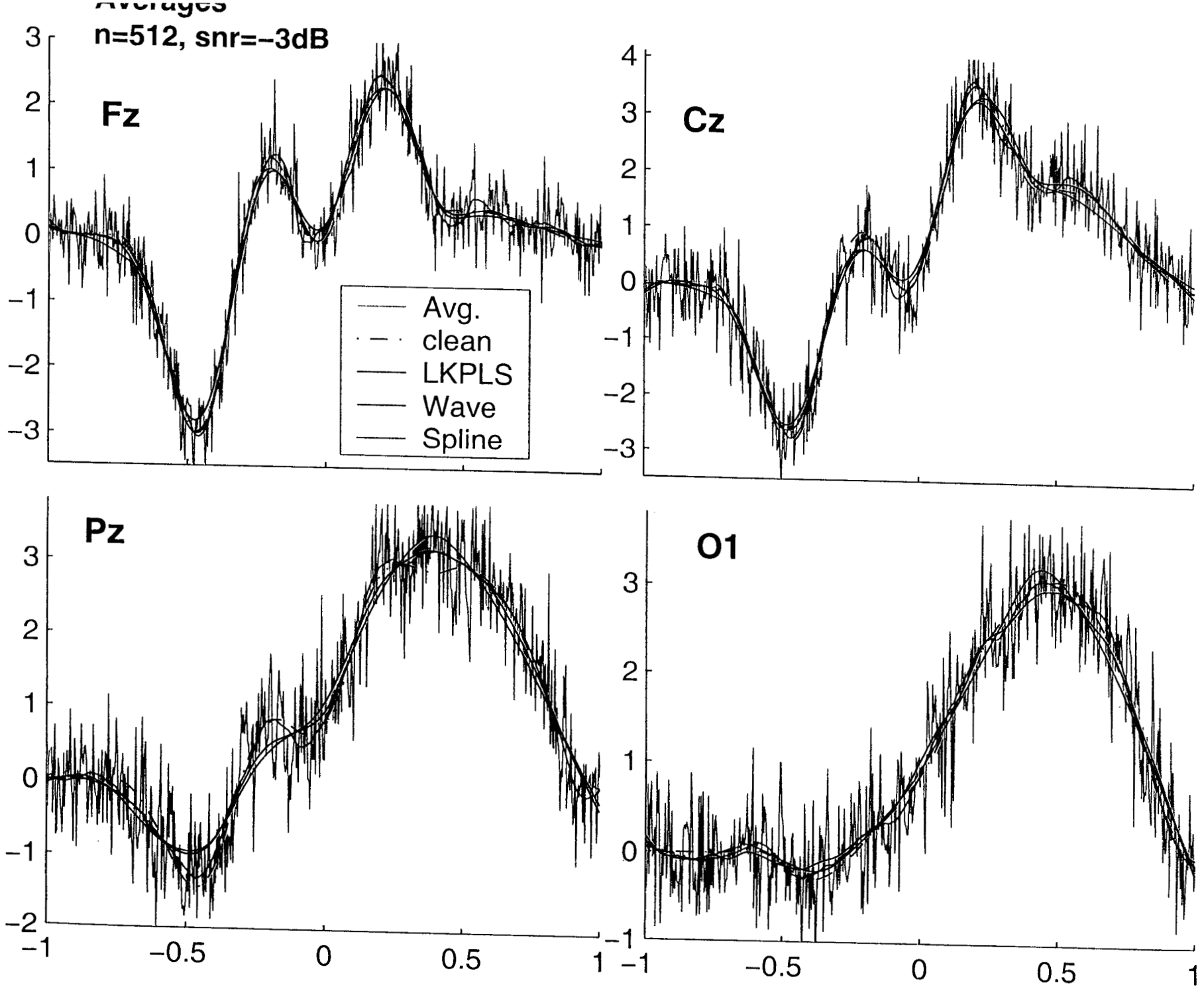
Noise	SNR 1dB		SNR 5dB		SNR 10dB		SNR 15dB	
	LKPLS	Wave	LKPLS	Wave	LKPLS	Wave	LKPLS	Wave
128	.22	.33	.18	.22	.13	.14	.10	.10
256	.17	.25	.13	.17	.10	.12	.08	.08
512	.12	.19	.10	.14	.09	.10	.07	.07
1024	.09	.15	.09	.11	.08	.08	.07	.05
2048	.	.12	.	.09	.	.06	.	.04

Table 2: Normalized root mean squared error.

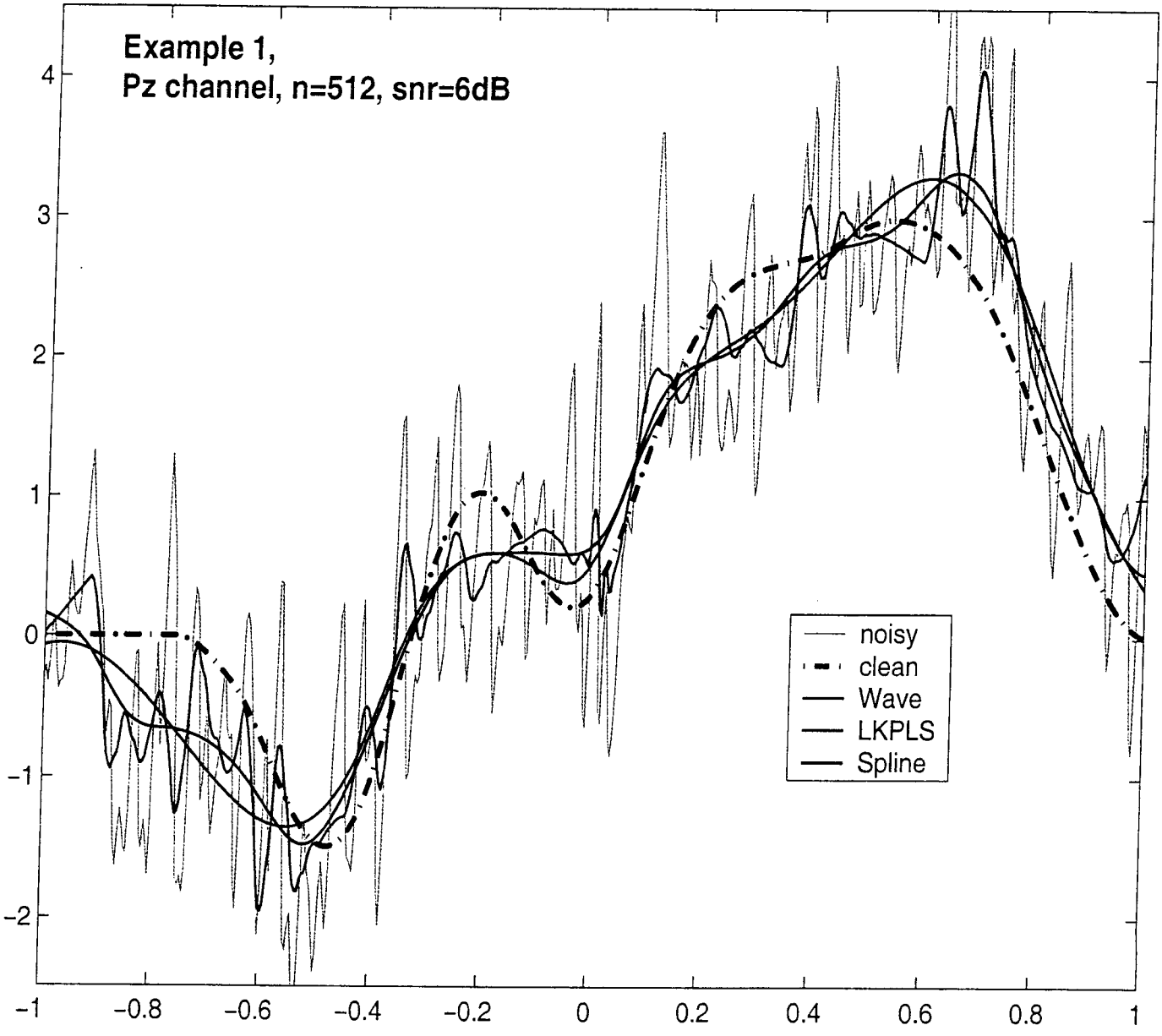
- ERP - white, Gaussian noise

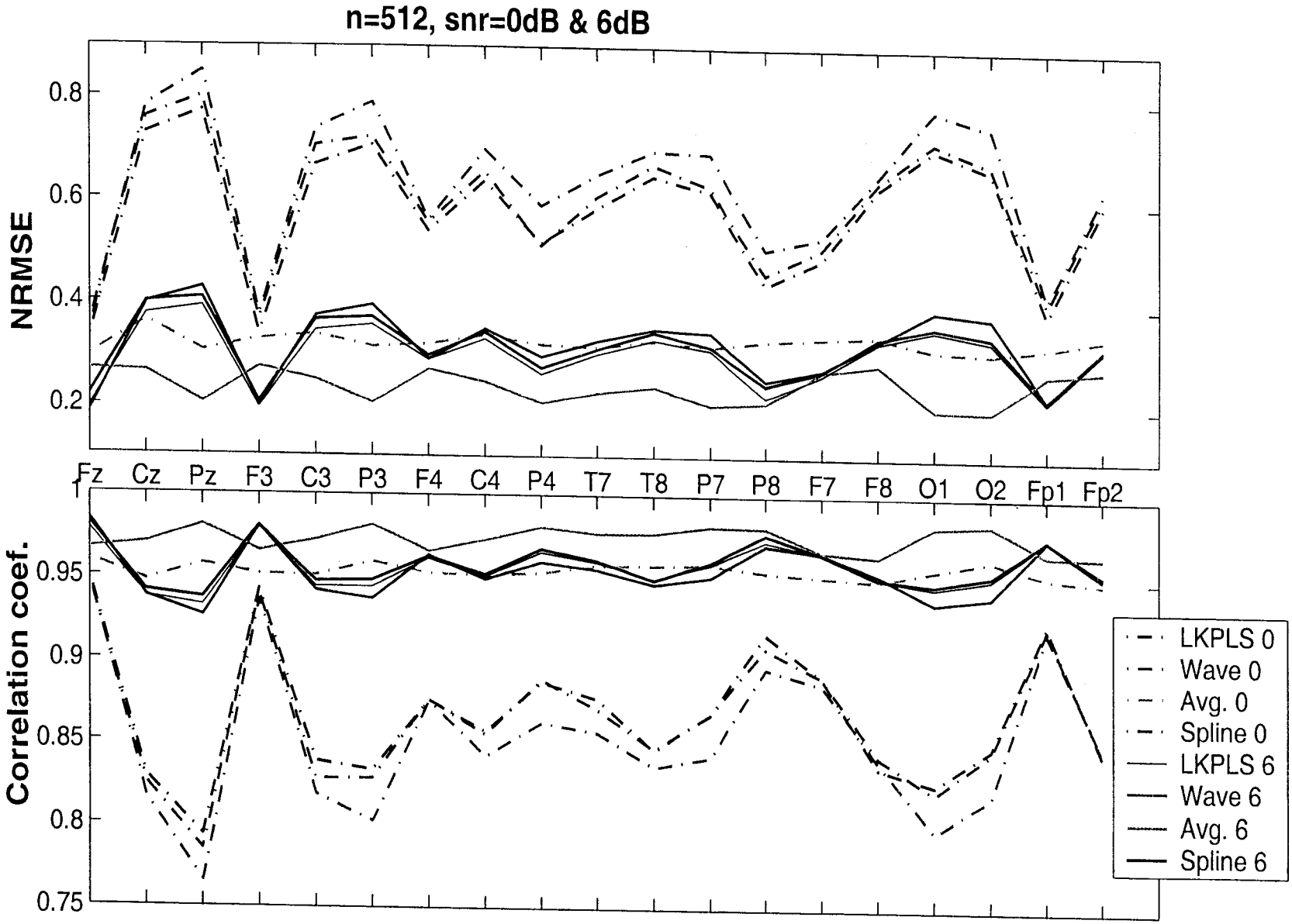


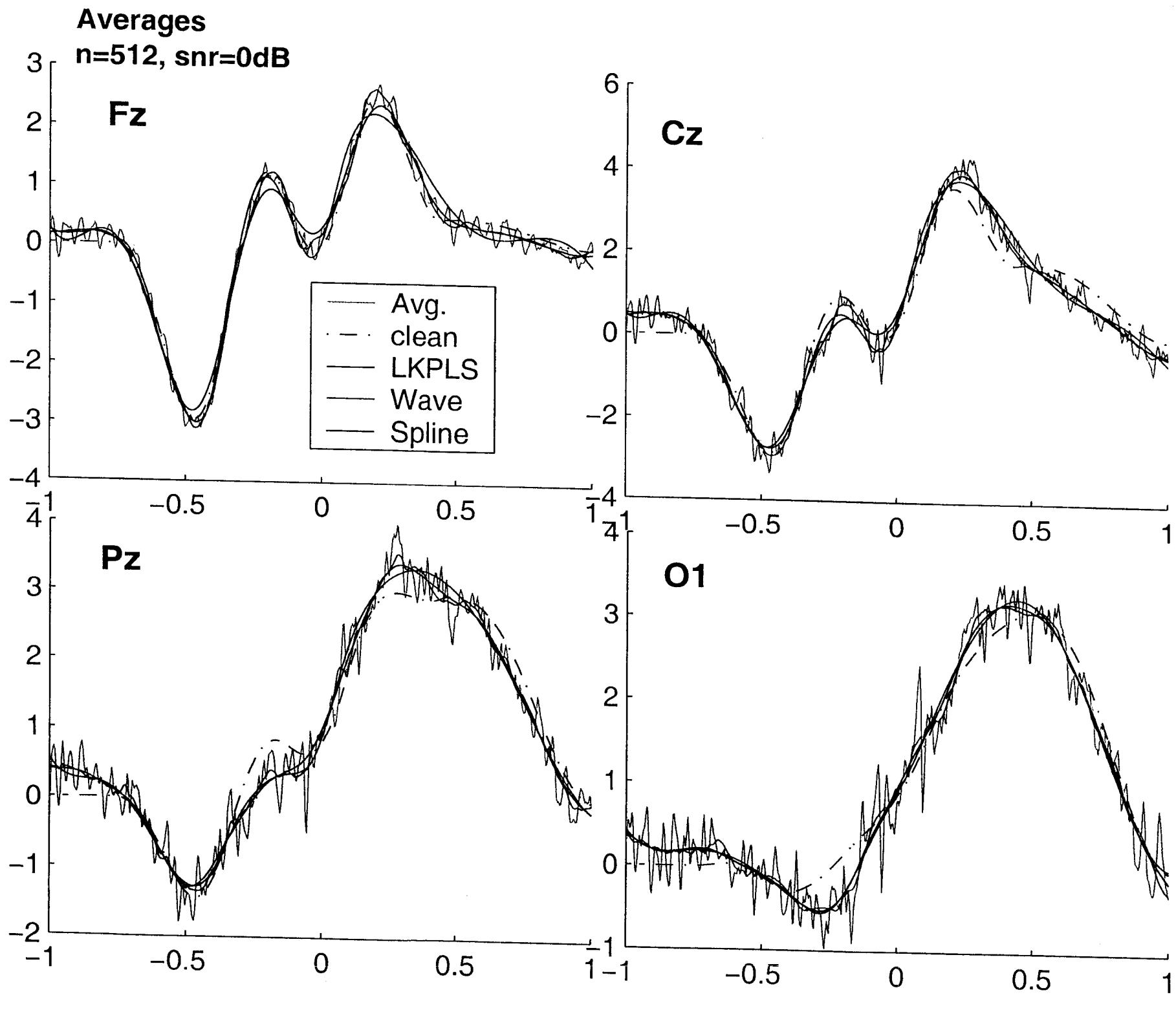

n=512, snr=–12dB & 6dB
NRMSE
1.2
1
0.8
0.6
0.4
0.2
0
Fz Cz Pz F3 C3 P3 F4 C4 P4 T7 T8 P7 P8 F7 F8 O1 O2 Fp1 Fp2
Correlation coef.
1
0.8
0.6
LKPLS 12
Wave 12
Avg. 12
Spline 12
LKPLS 6
Wave 6
Avg. 6
Spline 6


Averages
n=512, snr=−3dB
Fz
Cz
Pz
O1
Avg.
clean
LKPLS
Wave
Spline

- ERP - spontaneous EEG like noise




n=512, snr=0dB & 6dB
NRMSE
0.8
0.6
0.4
0.2
Fz Cz Pz F3 C3 P3 F4 C4 P4 T7 T8 P7 P8 F7 F8 O1 O2 Fp1 Fp2
Correlation coef.
1
0.95
0.9
0.85
0.8
0.75
LKPLS 0
Wave 0
Avg. 0
Spline 0
LKPLS 6
Wave 6
Avg. 6
Spline 6


Averages
n=512, snr=0dB
Fz
Cz
Pz
O1
Avg.
clean
LKPLS
Wave
Spline

Discussion, future work

- comparable results with existing state-of-the-art smoothing and de-noising techniques
- the construction of the (locally based) kernel PLS regression basis allows to incorporate the prior knowledge about the signal of interest
- input samplings dimensionality not crucial problem in (locally based) kernel PLS smoothing - e.g. images de-noising
- multivariate (locally based) kernel PLS allows straightforward extension to higher dimensional smoothing problems *plus* the existing correlation among the signals determine the basis construction - e.g. spatio-temporal smoothing of EEG recordings
- possibility to combine shrinkage and selection techniques *or* better model selection techniques ?
- computational disadvantages of kernel based approaches can be compensated by "segmentation" in the case of locally based kernel PLS ?
- smoothing real world biological signals - ERP, eye-blinks, etc.

References

1. Rosipal R., Trejo L.J.: Kernel Partial Least Squares Regression in Reproducing Kernel Hilbert Space. Journal of Machine Learning Research, 2:97–123, 2001.
2. Cherkassky V., Shao X., Mulier F.M., Vapnik V.N.: Model Complexity Control for Regression Using VC Generalization Bounds. IEEE Transaction on Neural Networks, 10:1075–1090, 1999.
3. Cherkassky V., Shao X.: Signal estimation and denoising using VC-theory. Neural Networks, 14:37–52, 2001.
4. Vapnik V.N.: Statistical Learning Theory. John Wiley & Sons, 1998.